Sesión 2: Mi Primera Condición de Carrera

Concurrencia

Ángel Herranz

Febrero 2019

Universidad Politécnica de Madrid

Concurrencia

 Simultaneidad

\+

Sincronización + Comunicación

Recordad

```
public static int x = 0;
```

```
public class Inc {
  public static void
    main(String args[]) {
    x = x + 1;
  }
}
```

```
public class Dec {
  public static void
    main(String args[]) {
    x = x - 1;
  }
}
```

Recordad e Implementad (10')

```
public static int x = 0;
```

```
public class Inc {
  public static void
    main(String args[]) {
    x = x + 1;
  }
}
```

```
public class Dec {
  public static void
    main(String args[]) {
    x = x - 1;
  }
}
```

- Implementarlo con lo que hemos aprendido

Acciones atómicas

```
$ javap -c Inc
public class Inc {
  ...
  public static void main(java.lang.String[]);
    Code:
       0: getstatic     #2                  // Field x:I
       3: iconst_1
       4: iadd
       5: putstatic     #2                  // Field x:I
       8: return
  ...
}
```

Procesos

- ...
- Podéis pensar en un proceso como en una CPU (con su memoria, su contador de programa, sus registros, su pila, etc.)

 Ejecutar a mano el programa

- ¿Cuál es el valor esperado de x?
- ¿Cuáles son los valores posibles ejecutando a mano el programa concurrente y teniendo en cuenta las acciones atómicas?
- ¿Coinciden?

Concepto: condición de carrera

Resultados indeseados por interacción de dos o más procesos

=

condición de carrera

Concepto: sección crítica

Porción de código que puede dar lugar a una condición de carrera

=

sección crítica

Secciones críticas

```
public class Inc {
  public static void
    main(String args[]) {
    x = x + 1;
  }
}
```

```
public class Dec {
  public static void
    main(String args[]) {
    x = x - 1;
  }
}
```